



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/825,249	04/02/2001	Rahul Sharma	SUNMP006	2037

25920 7590 01/17/2006

MARTINE PENILLA & GENCARELLA, LLP
710 LAKEWAY DRIVE
SUITE 200
SUNNYVALE, CA 94085

EXAMINER

PHAM, CHRYSTINE

ART UNIT	PAPER NUMBER
----------	--------------

2192

DATE MAILED: 01/17/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/825,249	Applicant(s) SHARMA ET AL.	
	Examiner Chrystine Pham	Art Unit 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 19 September 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1, 4-9, 13-21 and 25 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1, 4-9, 13-21, 25 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-352.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on September 19th 2005 has been entered.
2. This action is responsive to the Amendment filed on September 19th 2005. Claims 1, 4-5, 9, 13, 18, and 25 have been amended. Claims 2-3, 10-12, 22-24 have been canceled. Claims 1, 4-9, 13-21, and 25 are pending.

Response to Arguments

3. Applicants' arguments against the Nally reference have been considered but are moot in view of the new ground(s) of rejection.
4. Other arguments against the secondary references (e.g., Chung, Apte, and Savage) have been fully considered but they are not persuasive.

❖ With respect to Chung, Applicants contend, "C2 cites merely note possible **states** of a process (**volatile** and **persistent**), and are general as to **restoring** and **recovery**" (emphasis added)(starting page 12 of 14, last paragraph) and "**While these states are related to recovery**, none of these references to states described the claimed replication

of data, which is based on methods and apparatus which establish levels of importance of replicating data” (page 13 of 14, 1st full paragraph).

➤ The Examiner respectfully disagrees. First, the feature of “replication of data, based on methods and apparatus which establish levels of importance of replicating data” is not clearly established in the plain language of the claims. Merely cited as “the state management type being one of a recoverable state or a non-recoverable state, the recoverable state being one of a memory replicated state management type or a disk replicated state management type” (claim 1), it is not clear which state management type (i.e., disk-replicated or memory-replicated) is more important than which. Furthermore, the state management of the first type (e.g., recoverable state), as claimed, does not necessarily have a “level of importance” that is different from that of the state management of the second type (e.g., non-recoverable state) since, as claimed, there are only two possible state management types (i.e., memory replicated and disk replicated) and the recoverable state is claimed as being **either** memory replicated **or** disk replicated. Since the recoverable state management type is not claimed as being of an exclusive type (i.e., exclusively disk replicated, *or* exclusively memory replicated) to clearly distinguish it from the non-recoverable state management type, the two different management types are not deemed to have different “levels of importance”. Rather, as claimed, there is no clear distinction between which state objects (hence, entity bean objects) are to be memory replicated and which state objects are to be disk replicated since any recoverable state object can be replicated to memory or disk.

➤ Furthermore, as has been established in the final Office Action (page 8), Chung was relied upon to teach the features recited in original claims 6, 7, 9, and 25 that were missing from Nally, i.e., “the state management type identifying a **policy for migration** of a state object from one server process to another server process” (recited in original claims 6 and 25) and “managing checkpoints” (recited in original claim 7). As pointed out by Applicants, col.2:6-11 of Chung discloses different states of a process as volatile and persistent. Chung identifies the volatile state as including any process information that would normally be lost upon a failure. In col.2:60-67, Chung further discloses a **checkpoint** and restoration techniques for saving the process state (including persistent and volatile states) during normal execution, and thereafter **restore/recover** the **saved state** during a **recovery** mode following a failure. As has been established in the final Office Action, col.5:10-13 of Chung expressly discloses **migrating** the process to a remote processing node. In col.5:49-65, Chung further discloses storing (i.e., managing) a copy of the volatile state (i.e., recoverable state) in an area of nonvolatile memory, such as on disk 100, which may reside locally (i.e., memory replicated) or on a remote processing node. Clearly, Chung suggests replicating and migrating the state to a different server.

❖ With respect to claims 16-17, and 19-21, Applicants generally assert, “the Apte and the Savage references do not remedy the deficiencies as discussed regarding to Nally and Chung references” (page 13 of 14, last paragraph) without providing specific arguments against the disclosed features for which these references were relied upon, the

incorporation of these references into new grounds of rejection set forth below is considered proper and maintained.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

6. Claims 1, 4, and 5 are rejected under 35 U.S.C. 102(e) as being anticipated by Apte et al. of record (US 6269373 B1, *Apte et al.*).

As per claim 1, *Apte et al.* teach a method and system for partitioning container-managed state for a Java base application, comprising the operations of:

- classifying individual entity bean objects (see at least *EJB 1208* FIG.12 & associated text) with a particular modular state management type (see at least *1204 Container* FIG.12 & associated text), the state management type being one of a recoverable state or a non-recoverable state, the recoverable state being one of a memory replicated state management type or a disk replicated state management type (see at least *1222 Back-end storage* FIG.12 & associated text;);

- providing a plurality of modular state objects/state partitions, each state object storing a state of a corresponding entity bean object (see at least *1208 EJB*, *1210-1214*, *1206 Tie* FIG.12 & associated text) within a memory address space of a Java server process (see at least *1202 server* FIG.12 & associated text), wherein each state object is associated with the state management type of the corresponding entity bean object (see at least *1208*, *1206*, *1204* FIG.12 & associated text); and
- providing state management (see at least *1204 Container* Fig.12 & associated text) for each entity bean object (see at least *1208 EJB* Fig.12 & associated text) using a state object corresponding to the respective entity bean object (see at least *1202 Server*, *1204 Container*, *1206 Tie*, *1208 EJB* Fig.12 & associated text; *application's state information, container, persist references, other servers* col.1:40-col.2:5), the providing state management being based on the associated state management type that is associated with the state object corresponding to the respective entity bean object (see at least *1206 Tie*, *1204 Container*, *1208 EJB* FIG.12 & associated text) the providing state management comprising replicating each one of the plurality of state objects is replicated in a state server (see at least *1208*, *1206*, *1204* FIG.12 & associated text; *application's state information, container, persist references, other servers* col.1:40-col.2:5), the state server for a particular state object being dedicated to a state management type that corresponds to the state

management type that is associated with the particular state object (see at least *1202 Server, 1204 Container, 1206 Tie, 1208 EJB* Fig.12 & associated text).

As per claim 4, the rejection of base claim 1 is incorporated. *Apte et al.* further teach the operation of grouping the state objects based on the type of state management to which the corresponding entity bean object is classified (see at least *EJBs, container, protocol, persistence* col.7:25-55).

As per claim 5, *Apte et al.* teach the method as applied to claim 4, wherein the state management type (see at least *1204* FIG.12 & associated text) into which a group of state objects are grouped (see at least *1210-1214* FIG.12 & associated text) identifies a policy for replication of the group of state objects to the dedicated state server (see at least *1202* FIG.12 & associated text) that is dedicated to the particular state management type corresponding to the group (see at least *EJBs, protocol, particular server, mechanisms, persistence, container* col.7:25-55).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.

Patentability shall not be negated by the manner in which the invention was made.

8. Claims 6-7, 9, 13-20, and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Apte et al.* in view of *Chung et al.* of record (US 6105148, *Chung et al.*).

As per claim 6, *Apte et al.* teach the method as applied in to claim 4. *Apte et al.* do not expressly disclose the state management type identifying a policy for migration of a state object from one server process to another server process. However, *Chung et al.* teach the a method and system for providing different types of state management (e.g., see *volatile state 30 & persistent state 120* FIG.1 & associated text, col.2:6-11, col.5:53-60) for entity bean objects (e.g., FIG.8A & associated text) wherein checkpoints are managed using state objects (e.g., FIG.4 & associated text, col.2:62-66, col.4:50-55, col.8:1-3) and state management unit identifies a particular mechanism for recovery of states for entity bean objects (e.g., col.2:62-66, col.4:50-55), which are migration capable between server processes (e.g., FIG.2 & associated text, col.5:10-13). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Chung et al.* into that of *Apte et al.* which would produce the expected result with reasonable success. And the motivation for combining the teachings would have been that utilizing state objects in managing checkpoints enables the monitoring and persisting of the states as well as detection of data conflicts which might occur following each checkpointed state, thus, enforcing data consistency and allowing the recovery (based on the methods specified in the recovery mechanism) of the application process to it previous state. Furthermore, it would have been obvious to one

of ordinary skill in the pertinent art at the time the invention was made that specifying migration mechanism using state objects within state management units enables the application in the first processing server to be exported to, installed, and deployed on a second processing server in the event of permanent or long-term hardware failure of the first server (see at least *Chung et al.* col.5:5-13).

As per claim 7, the rejection of base claim 1 is incorporated. Claim recites limitations, which have been addressed in claim 6, therefore, is rejected for the same reasons as cited in claim 6.

As per claim 9, *Apte et al.* teach a method for partitioning container-managed state for a Java application, comprising the operations of:

Partitioning individual entity bean objects of the Java application into state partitions, wherein the state partitions manage concurrency for the Java application (see at least *EJBs, mechanisms, concurrency, behavior, container* col.7:25-55), the partitioning being the state objects corresponding to the entity bean objects (see at least *1208, 1210-1214, 1206, 1204* FIG.12 & associated text);

Classifying individual entity bean objects within each state partition using state management units, wherein each state management unit is a collection of the state objects corresponding to one particular state management type for recoverable state of the respective corresponding entity bean objects (see at least *1208, 1210-1214, 1206, 1204* FIG.12 & associated text); and

Replicating each state management unit in one of a plurality of state servers (see at least *server 104* FIG.1 & associated text; *additional servers* col.3:58-col.4:5) according to the particular state management type that corresponds to the particular state objects classified in the state management unit (see at least *EJBs, protocol, particular server, mechanisms, persistence, container* col.7:25-55).

Apte et al. do not expressly disclose migration capable state of the respective corresponding entity bean objects. However, *Chung et al.* disclose managing migration capable state of the processes (see claim 6). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Chung et al.* into that of *Apte et al.* for the inclusion of migration capable state. And the motivation for doing so would have been the same as has been cited in claim 6.

As per claim 13, the rejection of base claim 9 is incorporated. *Apte et al.* further teach the operation of using a control module/repository (maintaining state partition specifications) to manage dynamic partitioning/replication of the state of the application via the state partitions and the state management units (see at least *container, mechanisms, concurrency* col.7:25-55).

As per claim 14, the rejection of base claim 13 is incorporated. *Apte et al.* further teach wherein the state partitions and state management units are modular (see at least *1208 EJB, 1210-1214, 1206 Tie, 1204 Container* FIG.12 & associated text).

As per claim 15, the rejection of base claim 14 is incorporated. *Apte et al.* further teach wherein additional state management types for the state management units can be defined (see at least *1208 EJB*, *1210-1214*, *1206 Tie*, *1204 Container*, *1202 Server* FIG.12 & associated text; col.3:58-col.4:5; *EJBs*, *server*, *protocol* col.7:25-55).

As per claim 16, *Apte et al.* teach the method as applied to claim 15. *Apte et al.* further disclose a method and system wherein each state partition serialize transactions for entity bean objects within a particular state partition (e.g., col.15:21-27, col.15:67-col.16:5, col.16:57-65). *Apte et al.* further disclose entity bean objects (e.g., see *EJB 1208* FIG.12 & associated text) of the application are partitioned into state partitions during pre-deployment (e.g., see Abstract, *fields 1210-1214* FIG.12 & associated text).

As per claim 17, the rejection of base claim 16 is incorporated. *Apte et al.* further teach each state partition allows only one concurrent transaction to be performed on the entity bean objects within the particular state partition during a given time period (see at least *container*, *mechanisms*, *concurrency* col.7:25-55).

As per claim 18, *Apte et al.* teach a system application for partitioning managed container-managed state for a Java based application (see at least *704*, *728* FIG.7 & associated text), comprising:

- an application having a plurality of entity bean objects (see at least *1208* Fig.12 & associated text), each entity bean object comprising a state

- management type (see at least 1204 FIG.12 & associated text), the state management type being one of a recoverable state or non-recoverable state (see at least 1222 FIG.12 & associated text), the recoverable state being one of a memory replicated state management type or a disk replicated state management type (see at least 1108, 1112 FIG.11 & associated text);
- a plurality of state objects (see at least 1210-1214 FIG.12 & associated text), each state object storing a state of a corresponding entity bean object (see at least 1208 FIG.12 & associated text) within a memory address space of a Java server process (see at least 1202 FIG.12 & associated text), wherein each state object is associated with a particular state management type of the corresponding entity bean object (see at least 1210-1214, 1208, 1206, 1204, 1202 FIG.12 & associated text); and
 - a plurality of state management units (see at least *additional servers* col.3:55-col.4:5) that classify the state objects, a particular state object being classified into a particular state management unit based on the particular state management type of the corresponding entity bean object wherein the state management units facilitate state management for each entity bean object;
 - a state server dedicated to each state management type, the state management type identifying a policy for replication of a state object to a state server dedicated to a particular state management type (see at least *EJBs, protocol, particular server, mechanisms, persistence, container* col.7:25-55); and

- a replicated state manager configured to replicate a particular state management unit to the state server that is dedicated to the particular state management type of the particular state object that is classified into the particular state management unit to be replicated (see at least *EJBs, protocol, particular server, mechanisms, persistence, container* col.7:25-55).

Apte et al. do not expressly disclose a policy for migration of a state object from one server process to another server process. However, *Chung et al.* disclose a policy for migration of a state object from one server process to another server process (see claim 6). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Chung et al.* into that of *Apte et al.* for the inclusion of a policy for migration of a state object from one server process to another server process. And the motivation for doing so would have been the same as has been cited for claim 6.

As per claim 19, see claim 16.

As per claim 20, see claim 13.

As per claim 25, see claim 6.

9. Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Apte et al.* in view of Nally et al. of record (US 6298478 B1, *Nally et al.*).

As per claim 8, the rejection of base claim 1 is incorporated. *Apte et al.* do not expressly disclose the operation of performing lock management using the state objects. However, *Nally et al.* teach the operation of performing lock management using the state objects (e.g., *transaction isolation, instances, EJB col.3:49-col.4:43*). *Apte et al.* and *Nally et al.* are analogous art because they are both directed to persisting state information for EJBs. It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Nally et al.* into that of *Apte et al.* for the inclusion of performing lock management using the state objects. And the motivation for doing so would have been to avoid the performance penalties inherent in the conventional lock management using serialization, thus enables multiple concurrent transactions/accesses to the same entity bean object while ensuring consistency and independent views among the different transactions (see at least *Nally et al. col.3:45-col.4:45*).

10. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Apte et al.* in view of *Chung et al.* further in view of *Savage et al.* of record (US 6604110, *Savage et al.*).

As per claim 21, *Apte et al.* teach the system as applied to claim 20 wherein the repository manages replication of the state of the Java application during runtime (e.g., see claim 13). *Apte et al.* do not expressly disclose the repository manages migration of state of the Java application. However, *Savage et al.* disclose a repository (e.g., see

generic metadata repository 200 FIG.13 & associated text) managing migration of enterprise application data (e.g., see *generate migration specifications 202* FIG.13 & associated text, col.1:22-25 & 52-56, col.21:1-6). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Savage et al.* into that of *Apte et al.* which would produce the expected result with reasonable success. And the motivation for combining the teachings would have been that a repository, which specifies migration protocol, enables the source application data (e.g., properties, fields, states) persisted in the repository of one operational system to be analyzed in order to generate metadata/migration protocol which would specify how the data on that particular operational system are logically transformed (or made independent) from the underlying operational system model to other logical and physical structure of data warehouses (aligning with target business/enterprise structures) on other operational systems so that said data can be logically mapped, cross-referenced, or incorporated into diverse type business/enterprise applications.

Conclusion


11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chrystine Pham whose telephone number is 571-272-3702. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

CP
January 9, 2006



TUAN DAM
SUPERVISORY PATENT EXAMINER